DOCUMENT RESUME

ED 148 369 IR 005 478

AUTHOR Chandler, John S.; DeLutis, Thomas G.
TITLE A Methodology for the Performance Evaluation of Information Systems Under Multiple Criteria.
PUB DATE 76
NOTE 18p.

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS Evaluation; *Evaluation Methods; *Information Systems; Information Utilization; *Interaction Process Analysis; Man Machine Systems; Needs; *Performance Criteria; Problem Solving; Systems Analysis

ABSTRACT

An information system can be viewed as a symbiotic relation between the users of the system and the system's hardware, software, and data resources. Ideally, performance evaluation of these systems should be based on the achievement of user based criteria while concurrently minimizing resource costs. Current evaluation technologies concentrate on either the user or the resources. This paper presents a methodology that addresses this problem of multiple criteria evaluation of information systems. A set of evaluative procedures is described which provides the analyst with the ability to identify a system's critical resources relative to the attainment of user oriented performance objectives, and which identifies changes in their behavior in order to minimize the non-achievement of these goals. The methodology has no inherent restrictions with regard to the complexity of the system's hardware and software resources or to the services provided to the user, and is applicable to both the design process and to the analysis of production systems. (Author/VT)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED148369

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

A METHODOLOGY FOR THE PERFORMANCE EVALUATION OF INFORMATION SYSTEMS UNDER MULTIPLE CRITERIA

John S. Chandler and Dr. Thomas G. DeLutis

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

John S. Chandler

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) AND USERS OF THE ERIC SYSTEM."

IR 005 478

A Methodology for the Performance Evaluation of Information Systems Under Multiple Criteria

John S. Chandler and Dr. Thomas G. DeLutis
Department of Computer and Information Science
The Ohio State University
Columbus, Ohio

ABSTRACT

An information system can be viewed as a symbiotic relation between the users of the system and the system's hardware, software and data resources. Ideally, performance evaluation of these systems should be based on the achievement of user based criteria while concurrently minimizing resource costs. Current evaluation technologies concentrate on only one aspect of the system, i.e., either the user or the resources. The ability to ascertain the impact of resource modification on user goals or vice versa is not readily achievable with these technologies.

This paper presents a methodology that addresses this problem of multiple criteria evaluation of information systems. A set of evaluative procedures are described which provide the analyst with the ability to identify a system's critical resources relative to the attainment of user oriented performance objectives and which identifies changes in their behavior in order to minimize the non-achievement of these goals. The methodology is a general one; it has no inherent restrictions with regard to the complexity of the system's hardware and software resources or to the services provided to the user and is applicable to both the design process and to the analysis of production systems.

I. INTRODUCTION

A computer based information system can be viewed as a symbiotic relationship between the users of the system and the system's hardware, software and data resources. Ideally, the system performs optimally when it achieves its user oriented objectives at minimum expenditure of system resources. One purpose of computer measurement is to accumulate statistics on selected resource usage during operation for the purpose of identifying ways in which systems throughput, response time and/or costs can be altered. Hopefully, the objective of these measurement activities is to make the system more responsive to the user environment or to retain its current behavior at reduced costs, or both.

The complexity of the problem is substantially increased when the system has multiple users (uses), each with differing performance goals. Consequently, it is quite possible to improve the system's performance with respect to one or more users (uses), at the expense of other users (uses). What is needed are evaluative techniques which provide the analyst with a capability to evaluate simultaneously the effects of modifying system resources on all system users. Furthermore, these technologies should guide the analyst in effecting system performance improvements vis-a-vis the system's throughput, response time, or cost.

The purpose of this paper is to present an evaluative technique for addressing the problem of multiple criteria performance evaluation of information systems. The goal of this technique is to provide the analyst with an evaluative procedure which:

1. ascertains the degree to which the system has attained the various user goals,
2. relates the goals of individual user classes to measurable system parameters,
3. identifies those information system activities which should be modified to improve system performance, relative to those goals.

The basis for the technique is an extension of the linear programming methodology commonly refered to as multiple goal programming. Furthermore, as will be evident, this technique can be viewed as a data analysis procedure to be used in conjunction with performance monitoring data.

II. THE CURRENT STATE OF PERFORMANCE EVALUATION

In March of 1973, ACM and NBS sponsored a Workshop on computer performance evaluation. One of the major results of that Workshop was a concensus that there have been two separate approaches to the evaluation of information system performance -- one which focuses on the computer system domain and the other whose attention is directed at the application system (user) domain (BOE75). Each have their own goals and measures: The computer system domain controls expenditures through resource utilization and the user domain evaluates the services provided via measures such as throughput and response time with respect to each service. Many participants in the Workshop indicated that most performance evaluation efforts fail because of a lack of understanding of these two distinct domains or a failure to account for them in either their evaluative methodologies or their analyses of measurement data. The Workshop also concluded that any performance evaluation activity "should recognize both the costs of a computer installation and the needs of users for service" (pg. 10, BOE75).

User oriented analyses with objective functions based on response time, throughput and cost have been (and are continuing to be) reported in the literature. Most frequently, queueing models are used as the basis of the methodologies because of the stochastic behavior of user arrivals (GAV76, BUZ73, NEI76 are representative of this type of analysis). Because of the necessity to maintain tractable models, the complexities of different users having dissimilar resource requirements are ignored or are viewed with mathematical niceties required for a model's analytical solution. As a result of these mathematical simplifications, the utilization and queueing statistics produced from these models fail in many cases to identify complex interactions between users and resources. The use of model results are, therefore, suspect when used to predict the impact on system performance resulting from the modification of a system's service characteristics or from the alteration of its exogenous loading. However, many benefits do arise from these models including closed form solutions and ready employment of measurement data.

Alternatively, performance evaluations can be made from the system's standpoint, treating the user and his goals in the aggregate. Again, there have been enumerable

reports of this approach in the literature. They can be categorized as subsystem studies and total system studies. In the former, a particular part of the information system complex is isolated with the subsystem user(s) represented by a stochastic generator. The most emphasized areas of research has been the I/O subsystem (ABA68, NAH73, HEL74) and CPU utilization (KLE72, LEW71). The problem with this level of evaluation is that, although providing valuable local intuitive insight, these models rarely relate to the ultimate information system user, and therefore do not provide realistic insight into global performance. Again, an advantage of these models is their ready use of measurement data for model validation.

Examinations of complete systems have also been made. Models of information systems, including the user component, have been built and/or exhaustive hardware/software measurements have been analyzed by Cantrell and Ellison, Gonzales and Kimbleton (CAN68, GON76, KIM72). Although results of these models include resource utilization statistics and user oriented measures such as response time, there is little attempt in these models to relate particular resource usage to the effect on user goal attainment. The measurements of user related goals are treated more as indices of global system performance than as performance indicators themselves. The user is pictured as a passive element, without substantive goals and unable to directly control resource activity or respond to lack of goal attainment. (An exception is David Linday's study of the KRONOS systems (LIN76).) But from practice it is evident that the user does indeed respond to the system's performance with respect to his needs and can alter resource usage. In fact, Buzen (BUZ76) has proposed some fundamental laws for computer performance which relate resource activity to global system/user measures. However, his reported research indicates no attempt at differentiating between user classes. Furthermore, to date, there is no documented empirical evidence to substantiate his laws.

The crux of the problem, then, is to establish a causal relationship between user goal attainment and system resource expenditures. The evaluative technique presented here is designed to establish such relationships and will be shown capable of achieving heretofore hard to obtain evaluative information. The methodology measures the impact of individual user classes on internal system performance and identifies system bottlenecks which inhibit the attainment of user goals. This is achieved by maintaining resource utilization statistics on a user class basis. The methodology presented here is an iterative one. The output of the evaluative procedure is a set of performance indicators which are to be used by the system analyst as guidelines for altering the current system in order to more closely satisfy user goals or reduce resource inefficiencies.

III. EVALUATIVE REQUIREMENTS

For the purposes of this discussion, an information system is viewed as being comprised of users, services, activities and goals as illustrated in Figure 1. It is assumed that one purpose of an evaluative procedure is to recognize and classify the system users according to the performance constraints imposed upon the system (i.e., goals) when honoring their requests. It is also assumed that an evaluative technology must identify the critical activities associated with the information system which impact system performance. Obviously, the complexity of the problem is increased substantially when a system supports diverse users or provides a wide spectrum of services. Whether or not the system is complex or simple,

the criteria for identifying system activities must be based upon the sensitivity of the system's performance to changes in activity behavior.

Information system activities, for the purpose of this evaluative procedure, are viewed as being a series of distinct yet interconnected subservices which are invoked during the servicing of a stream of user requests. Again, Figure 1 illustrates this view of a system. It must be emphasized that activities are most likely to be aggregations of one or more traditional computer system functions which perform the following tasks:

1. request (job) scheduling,
2. task management,
3. resource allocation,
4. secondary storage I/O processing, and
5. application processing.

It must be stressed that the choice of what constitutes an activity is part of the art of performance evaluation. However, a necessary condition for their selection is that they be measurable and that these measurements distinguish the service rates for separate classes of system users or services. It is also assumed that the role of performance measurement is to determine the current service rate for the jth activity with respect to the jth service (here identified by the variable $R_j(i)$).

Each activity is viewed within the methodology as an individual queueing system (illustrated in Figure 2) characterized as one or more priority queues and one or more identical servers. An activity's performance is treated in the aggregate. Additionally, the performance of the activity in servicing a request type is the sum of both the queue transit and service functions of the activity. For the purposes of this paper, the variable $R_j(i)$ is employed to identify this performance of activity Aj, with respect to service S_i. Also associated with activities are performance factors, βj's. which are the variables of the evaluative procedure. The interpretation of the β's is the change in the $R_j(i)$'s which should be effected in order to obtain satisfied system performance based on the current behavior of the system.

It should be noted that the problem of identifying a more appropriate level of activity performance, i.e., determining the appropriate values of the β's, is compounded by the multiple use of the activity by different and possibly conflicting services. Thus, the modification of activity's service rate to achieve one goal may be counter productive to the obtainment of another goal. The trivial solution to this problem is to replicate all the common activities until each service has its own unique sequence of activities. This solution, obviously, is not the least cost solution and therefore will very likely be rejected as being infeasible in all but very special cases. One goal of the methodology is to recognize the interdependencies and tradeoffs between multiple invocations of an activity by different services. Therefore, the objective of the methodology is to provide the analyst with the set of β's which minimizes the discrepancy between the system's performance and the user's requirements.

IV. A FUNCTIONAL VIEW

A search of evaluative technologies suitable to the problem described in the preceeding section has led to multiple goal programming as a foundation for the methodology. Briefly, multiple goal programming has the following formulation:

Minimize $\vec{P} \cdot \vec{D}$

Subject to $A\vec{x} + \vec{D} = \vec{G}$

where

$\vec{G}$ = the vector of goals

$\vec{D}$ = discrepancies from these goals

$\vec{P}$ = the penalties associated with the discrepancies

$\vec{x}$ = the level (or units) of system resource allocation, and

A = a matrix of technological coefficients which can be thought of as the rate at which the ith resource is used by the system in performing the jth type of service.

and where the objective function is to minimize the discrepancies and associated penalties. The solution to a multiple goal programming problem represents the best set of levels for the resource allocation vector, x, such that the objective function is minimized.

Goal programming was first formulated by Charnes and Cooper (CHA61) in 1961 to solve linear programming problems that had conflicting constraints. Ijiri (IJI65), developed the details of the procedure within the framework of mathematical programming. This technique has been used to solve problems in the areas of strategic management planning such as accounting control (Ijiri), advertising-media planning (Charnes and Cooper, (CHA68)), and resource allocation (Lee, (LEE72)). The employment of goal programming in conjunction with information system performance evaluation is a new use of the methodology.

Figure 3 illustrates the relationship between goal programming, the information system activities, and its servicing of user requests. The servicing of a request of type i is a sequence of activities, A_1, A_2, ... A_n. Associated with each activity A_j is assumed to be a measurable performance factor which represents the rate of service for the activity relative to the ith type of service. For the purposes of this discussion these measures are assumed to be the average time $R_j(i)$, for the activity to service a request of type j relative to service type i, and as illustrated in Figure 3, it is the sum of the queueing time and service time for the activity. In general, the measure could be a function $M(R_j(i))$ of the service time for the activity. In general, the measure could be a function $M(R_j(i))$ of the service time (formulation (F) below incorporates this concept).

The performance of the information system for service type i is given by the relation

$$T_S(i) = \sum_{j=1}^{n} R_j(i) \qquad (1)$$

where $T_S(i)$ is the average system response time to service requests of type i.

Assuming that the performance goal for the service is $T_G(i)$, then the discrepancies between performance and goal is given by

$$D(i) = T_G(i) - T_S(i).$$

Again, the general formulation of the goal could be a function M of the response time, say

$$G(i) = M(T_G(i)).$$

The purpose of any analysis procedure is to minimize the discrepancy D(i) between performance and goal. Essentially, this can be viewed as determining new performance levels for each activity in such a manner that the discrepancy D(i) is minimized (hopefully to zero). Letting β_j be a scaling factor to be applied to the jth activity, then the new performance level for the activity is $R_j(i)\beta_j$. Incorporating the β_j's into equation (1) and substituting (1) into (2) results in the following expression for the discrepancies:

$$D(i) = T_G(i) - \sum_{j=1}^{n} R_j(i)\beta_j$$

Goal programming's role in the methodology is to set the values for the β_j's such that the sum of the discrepancies are minimized. Figure 3 illustrates the effect of the β's on overall performance. Therefore, the formulation of the goal programming problem is:

$$\min \quad D(1) + \ldots + D(L)$$

$$\text{s.t.} \quad R_1(1)\beta_1 + R_2(1)\beta_2 + \ldots + R_N(1)\beta_N + D(1) = T_G(1)$$

$$\vdots$$

$$R_1(L)\beta_1 + R_2(L)\beta_2 + \ldots + R_N(L)\beta_N + D(L) = T_G(L)$$

or in matrix notation

$$\text{Min} \quad D(1) + \ldots + D(L)$$

$$\text{s.t.} \quad R\vec{\beta} + \vec{D} = \vec{T}_G$$

Discrepant performance relative to a goal can be a result of either over-achievement or under-achievement of that goal by the system. To accommodate this situation, multiple goal programming allows for discrepancies to be either negative, $D(i)^-$, or positive, $D(i)^+$. The objective of modifying the system performance for a given goal $T_G(i)$, is to drive both $D(i)^-$ and $D(i)^+$ to zero. If a goal is unattained, however, only one of these variables will be non-zero, since a goal cannot be both over and under achieved at the same time. If $T_G(i)$ is over-achieved, then

$$\sum_{j=1}^{n} (R_j(i)\,\beta_j) - T(i) > 0,$$

which, then in terms of positive and negative discrepancies, implies that $D(i) > 0$ and $D(i) = 0$. Conversely, if $T_G(i)$ is under achieved, then

$$\sum_{j=1}^{n} (R_j(i)\,\beta_j) - T_G(i) < 0,$$

which implies that $D(i) > 0$ while $D(i) = 0$. Combining these two relations yields the general goal constraint equation,

$$\Sigma_{j=1}^{n} (R_j(i) \quad _j) + D(i)^- - D(i)^+ = T_G(i).$$

Thus, the general model formulation in (C) becomes

$$\text{Minimize} \quad \Sigma_{i=1}^{L} (D(i)^+ + D(i)^-)$$

$$\text{s.t.} \quad R\vec{\beta} + I\vec{D}^- - I\vec{D}^+ = \vec{T}_G$$

where

$\vec{D}^+$ is the array of positive discrepancies,

$\vec{D}^-$ is the array of negative discrepancies.

With each discrepancy, positive or negative, a penalty function, $P^{\pm}_i$, can be associated. This penalty function need not be symmetric, i.e., $P^+_i \neq P^-_i$. For example, a response time that is too slow (under-achieved) is usually more detrimental than a response time that is too fast (over-achieved), and, in this case, $P^-_i > P^+_i$. The value assigned to $P^{\pm}$ is determined from the user's priorities and opportunity costs. If the over (under) achievement of a goal is inconsequential to the evaluation of the system, then the associated penalty, $P^+_i(P^-_i) = 0$. Combining the concepts of discrepancies and their associated penalties, the goal formulation of the information system evaluation problem is now

$$\text{Minimize} \quad \sum_{i=1}^{L} (P^+_{i=1}D(i)^+ + P^-_i D(i)^-)$$

$$\text{s.t.} \quad R\vec{\beta} + I\vec{D}^- - I\vec{D}^+ = \vec{T}_G \qquad \text{(E)}$$

A generalized formulation is achieved via the following substitutions in (E):

1. Technological coefficients -- the matrix of usage rates R is replaced by a function M(R) and
2. Goals -- $\vec{G}$, a vector of time-related performance goals, is substituted for T_G, where $\vec{G} = M(T_G)$.

Hence, the general goal programming formulation information system evaluation is

$$\text{Min} \quad \sum_{i=1}^{L} (P^+_i D(i)^+ + P^-_i D(i)^-)$$

$$\text{s.t.} \quad M(R)\vec{\beta} + I\vec{D}^- - I\vec{D}^+ = \vec{G} \qquad \text{(F)}$$

With the general formulation (F), a two phase evaluative technology is possible. During Phase I, the system's performance is measured in order to obtain the values $R_j(i)$ or $M(R_j(i))$. Phase II analyzes the system based on these performance values and on the achievement of overall user based goals. Goal programming is employed to identify how the system should be modified in order to minimize discrepancies and associated penalties. The resulting values for the β's identify those activities that are critical to the performance of a service or services. If the value for a $\beta_j = 1$ then the service characteristics of activity j were adequate to satisfy all the user's criteria. If $\beta_j < 1$, this implies that the current service rate of activity A_j is insufficient to meet the system's needs. If $\beta_j > 1$ then the current service rate of activity A_j is faster than necessary and there exists the possibility of excess capacity. Assuming that an activity can be viewed as a queueing system, the analyst has three avenues of action when a $\beta_j \neq 1$. First, he can analyze the queue's dispatching discipline in order to increase queue throughput (or possibly to replace it with a simpler one of the $\beta_j > 1$). Second, he can alter the service rate characteristics of the servers. Each alternative has potential cost savings on expenditures.

The methodology is by nature an iterative one since changes in the current configuration will cause the values for the $R_j(i)$'s (or $M(R_j(i))$'s) to be altered. Therefore, an interpretation of the resulting β's from one iteration does not guarantee optimal performance, but instead, provide a set of guidelines for the analyst to follow in performance improvement. Once modifications have been made to the system, the evaluative process must be repeated, i.e., new measurements must be obtained and goal attainment analyzed with the resulting set of new values for the β's. The objective of each iteration of the evaluative process and subsequent system modifications is to produce a sequence of β's each with successively less variance from its predecessor; that is, to cause a damping phenomenon to occur in order that the system settles into a configuration which minimizes the non achievement of its desired goals.

V. RELATIONSHIP TO DATA MEASUREMENT

As can be seen from the above discussion, the essence of the methodology are the values for the technological coefficients, $R_j(i)$. These can be obtained in numerous ways depending upon the accuracy needed for analysis purposes. The methodology has identified what types of measurements are required as a function of the activities to be characterized, the services to be performed and the performance measures to be employed. Queueing and service time statistics for activities identifiable by type of service are essential. The scope of the definition of an activity depends not only on the level of evaluative detail desired, but also on the level of accuracy attainable and its associated costs.

A good starting point is to use education estimates for the $R_j(i)$'s in the first iteration of the goal programming problem and to perform sensitivity analysis on these estimates once the goal programming solution has been obtained. Only those $R_j(i)$'s showing greatest sensitivity will require greater accuracy in value specification. Performance monitoring is a suitable technique for obtaining estimates for the $R_j(i)$'s when the system under investigation is an operational one. Both software and hardware monitors can be employed, the most appropriate depending upon the behavior characteristics of the $R_j(i)$'s. When the value for the $R_j(\cdot)$'s are constant or exhibit very small deviation for all the i's then aggregate statistics accumulated for an activity can be used effectively. Typical of this type of monitoring are fixed logic hardware monitors and job level software monitors. Conversely, when the $R_j(i)$'s show a wide spectrum of values (im-

plying intersecting and possibly conflicting user loads) then more sophisticated monitoring techniques are required. In this situation, the performance measurements must be identifiable by user class (or application). Hybrid monitors appear to be best suited for gathering the needed statistics.

In conclusion, the described methodology provides a technique for translating monitoring data into a measure of the system's performance vis-a-vis user (use) goals and provides a systematic method to identify design alternatives which will improve the system's performance when these goals have not been obtained.

REFERENCES

ABA68 Abate, J., Dubner, H., and Weinberg, S.B., "Queueing Analysis of the IBM 2314 Disk Storage Facility", JACM, vol. 15, no. 4, (1968), pp. 577-589.

BOE75 Boehm, B., and Bell, T.E., "Issues in Computer Performance Evaluation: Some Consensus, Some Divergence", PER, vol. 4, no. 3, (1975), pp. 4-39.

BUZ73 Buzen, J.P., "Computer Algorithms for Closed Queueing Networks with Exponential Services", CACM, vol. 16, no. 9, (1973), pp. 527-531.

CAN68 Cantrell, H.N., and Ellison, A.L., "Multiprogramming System Performance Measurement and Analysis", AFIPS Conf. Proc., vol. 22, (1968), pp. 213-221.

CHA61 Charnes, A., and Cooper, W.W., Management Models and Industrial Applications of Linear Programming, New York: John Wiley & Sons, Inc., 1961.

CHA68 Charnes, A., et al., "A Goal Programming Model for Media Planning", Management Science, vol. 14, no. 8, (April 1968), pp. 423-430.

GAV76 Gaver, D.P., and Hunfeld, G., "Multitype Multiprogramming: Probability Models and Numerical Procedures", Proc. of CPMME, (1976), pp. 38-42.

GON76 Gonzalez, G., "Using Covariance Analysis as an Aid to Interpret the Results of a Performance Measurement", Proc of CPMME, (1976), pp. 179-.86

HEL70 Hellerman, H.R., and Smith, H.J., "Throughput Analysis of Some Idealized Input, Output and Computer Overlap Configurations", Computing Surveys, vol.2,

IJI65 Ijiri, Y., Management Goals and Accounting for Control, Chicago: Rand McNally, 1965.

KIM72 Kimbleton, S.R., "The Role of Computer System Models in Performance Evaluation", CACM, vol, 15, no. 7, (1972), pp. 586-590.

KLE72 Kleinrock, L., and Muntz, R.R., "Processor-Sharing Queueing Models of Mixed Scheduling Disciplines for Time-Shared Systems", JACM, vol, 19, no. 3, (1972), pp. 464-482.

LEE72 Lee, S.H., Goal Programming for Decision Analysis, Philadelphia: Auerbach, 1972.

LEW71 Lewis, P.A.W., and Shedler, G.C., "A Cyclic-Queue Model of System Overhead in Multiprogrammed Computer Systems", JACM, vol. 18, no. 2, (1971), pp. 199-220.

LIN76 Lindsay, D.S., "A Hardware Monitor Study of a CDC KRONOS System", Proc. of CPMME, (1976), pp. 179-186.

NAH73 Nahourii, E., "Direct Access Device Simulation", IBM Systems Journal, vol. 13, no. 1, (1973), pp. 19-31.

NEI76 Neilson, J.E., "An Analytic Performance Model of Multiprogrammed Batch Time-Shared Computer", Proc. of CPMME, (1976), pp. 59-70.

SCH72 Schwetman, H.D., and Browne, J.C., "An Experimental Study of Computer Systems Performance", Proc. of ACM, (1972), pp. 693-703.

SUR76 "SIGMETRICS Survey - A Summary of Responses", PER, vol, 5, no.2 , (1976), pp. 4-39.

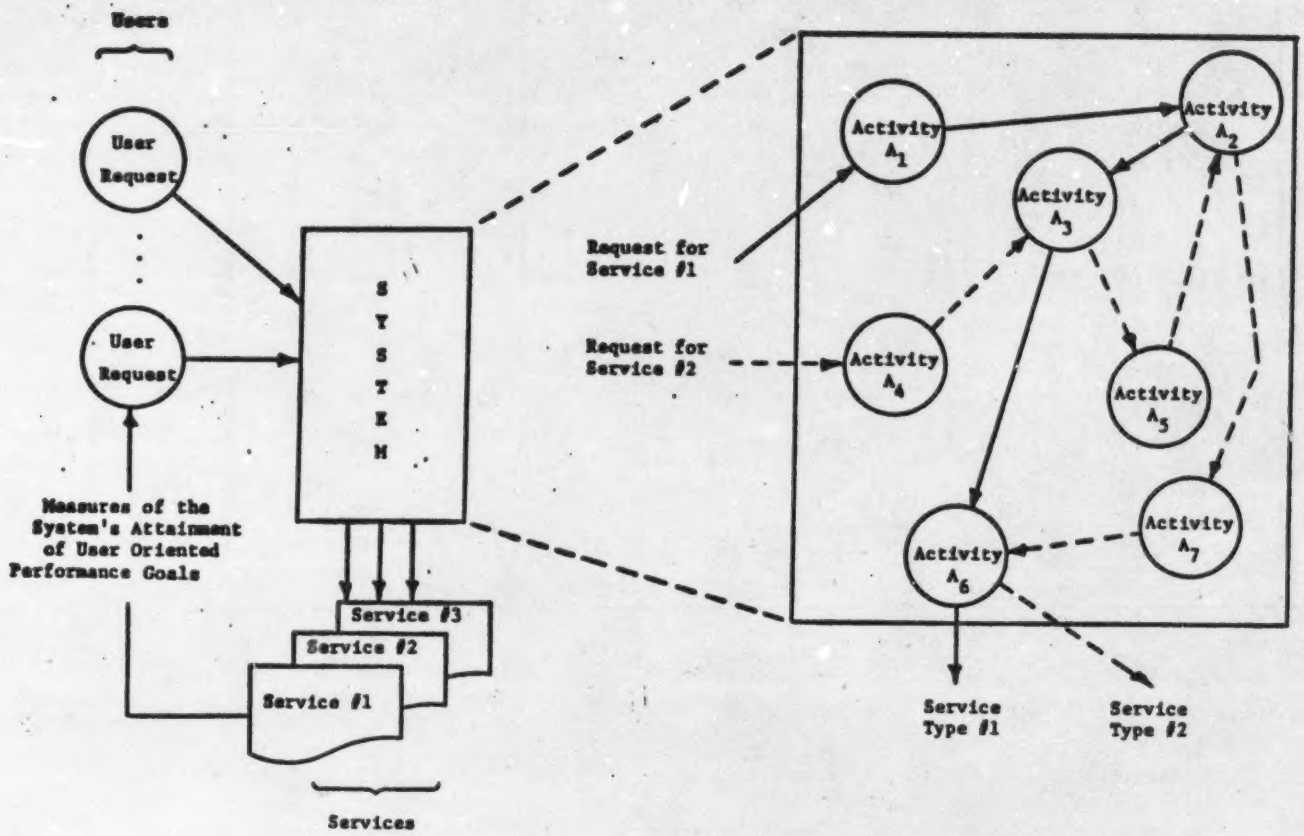


Figure 1. The Methodology's View of an Information System

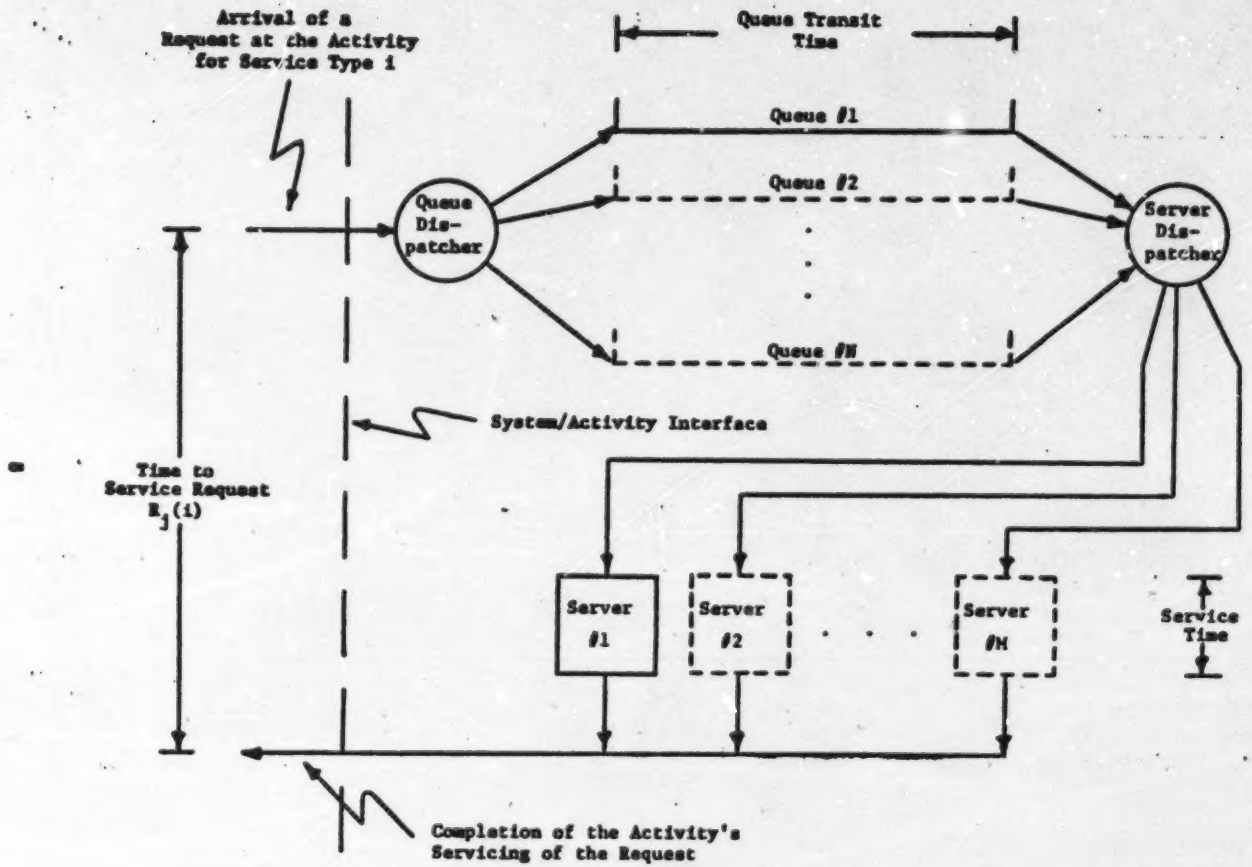
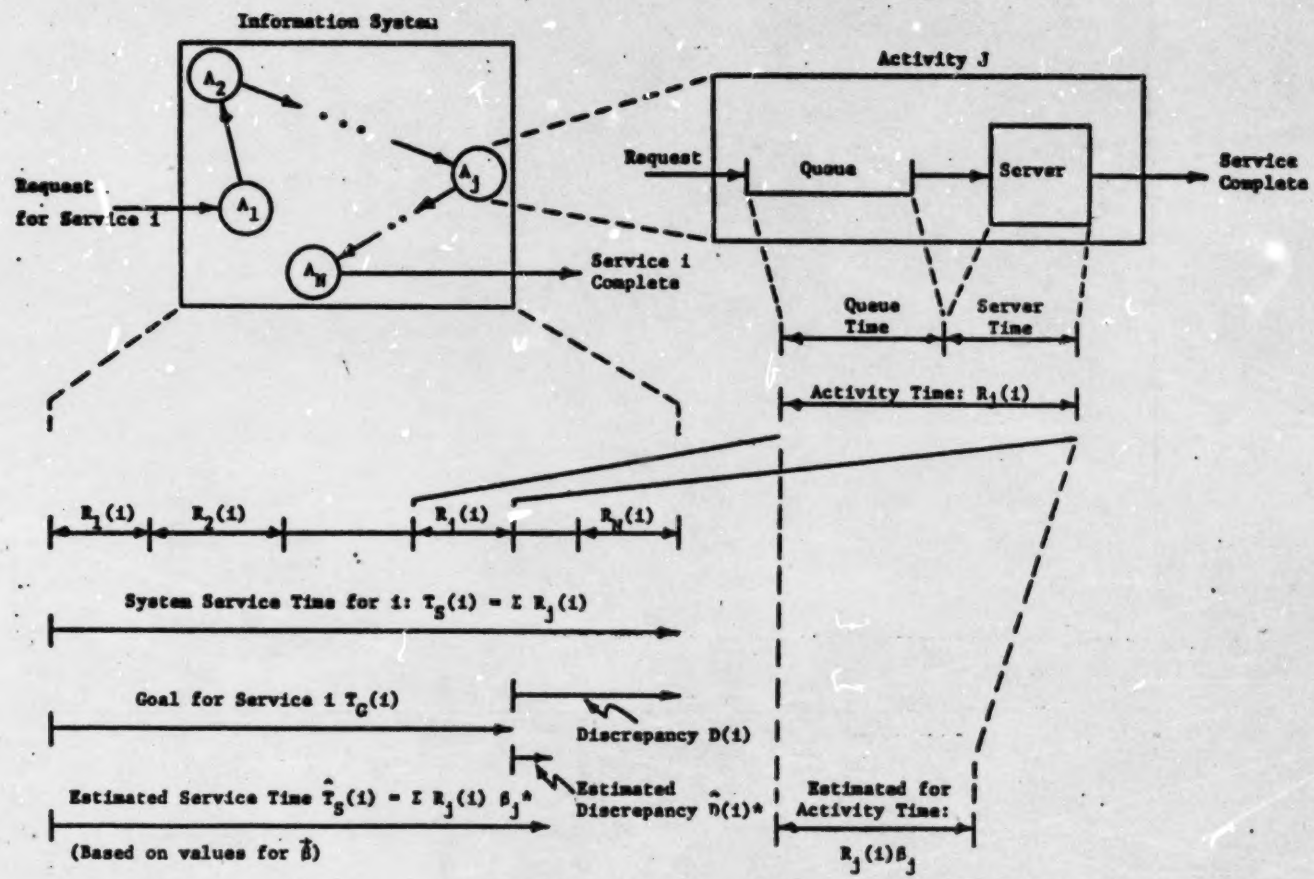


Figure 2. A Conceptual View of an Activity



*Values of β_j's and D(i)'s are the outputs of Goal Program Optimization Problem

Figure 3. Relationship Between Goal Programming and Information System Characterization